

Now you see it, now you don't: Understanding User Interface Visibility

Ian Michael Hosking, P. John Clarkson
University of Cambridge, Cambridge, UK
imh29@cam.ac.uk, pjc10@cam.ac.uk

Abstract

We live in a post-WIMP world. The traditional Windows, Icons, Menus and Pointers of the PC graphical user interface are no longer present in many hand-held devices. There has been a dramatic rise in the use of smart phones in particular, with Apple selling their billionth iPhone in 2016. This trend in devices and the shift to touch interfaces has caused concern with regards to usability and has been described by some as a “usability crisis”. This alleged crisis is born out of a proliferation of product features combined with a trend towards minimalisation in user interface style. This means that user interface functions are potentially becoming less visible. The challenge is to try and quantify and understand what is happening with regards to UI visibility, which is deemed to be a critical component creating a usable interface. This paper demonstrates an approach to determining a “visibility score” for a product’s user interface. The approach is applied to the home screen of an iPhone. This produces a visibility score of less than 10% in other words over 90% of the functions available are not visible at the top level. Such a score needs to be treated with caution but can help inform the general debate as well as creating useful insights for the designers of products.

Keywords: Usability, Visibility, WIMP, User Interfaces, User Experience

1 Introduction

We live in, what has been described as, a post-WIMP (Windows, Icons, Menu, Pointer) era [1-4], due to the growth of touch based devices in particular. Apple announced at the beginning of 2016 that they have 1 Billion active users on their non-WIMP products [5]. However, concerns have been raised about the usability of post-WIMP devices. Nielsen and Norman describe the situation as a “usability crisis” [6]. One of the key issues raised is the lack of visibility of user interface elements, which is deemed to be a critical and fundamental component of an effective user interface [6-8]. As Norman puts it, “The important design rule of a GUI is visibility” [8].

This alleged “crisis” is exacerbated by products that are suffering from a proliferation of features. For example, in Microsoft Word the number of commands in Word 1.0 was about 100 but by Word 2003 it had exceeded 1500. When Microsoft asked users what they wanted in the next version of Office, 9 out of 10 asked for features they already had in their current version [9]. Not only is not finding the required function a problem, but the opposite problem of accidentally activating an undesired function is a issue too. For example, Apple’s

iOS has a delete mode for apps that is activated by a 'touch and hold' on an app icon [10]. This can lead to accidental deletion of an app and its data or simply confusing the user. What we see is the combination of a proliferation of features with interfaces that lack visibility as they move to a minimalist design style. This is understandable as many post-WIMP devices must work within the constraints of smaller form factors such as those of smart phones. It is nonetheless a concern. The graphical user interface represented a major advance in user interface design from the command line interface style [11,12]. "Seeing and pointing" replaced "remembering and typing" [13-15]. But as we see a shift to touch interfaces are we potentially taking a retrograde step from "see and point" to "remember and swipe" or with non-touch gestural interfaces to "remember and wave".

This paper shows an approach to quantifying how many features are available and how many are visible directly to the user. Effectively this creates a simple "visibility score" for a user interface. This approach is applied to the home screen of Apple iOS on an iPhone smart phone. The pros and cons of such an approach are then discussed and recommendations made about how the approach can be applied and developed further.

2 Method

The Apple mobile and tablet operating system, iOS was chosen because it is widely used [5] and mature, being in its tenth version [10]. The device chosen was an iPhone7 (see figure 1), as the latest in a series of iPhones, which over a billion of have been made [16]. The multitude and diversity of factory installed and user chosen apps led to selecting "app launching and management", that is performed via home screens (see figure 2), as the target area for analysis. Note that Apple calls each subsequent 'screen' of apps an "additional home screen", hence the use of the plural term [10]. Out of the box there are two home screens. In addition to the ability to launch and manage apps, the home screens provide shortcuts to various actions for example turning the torch on or launching the camera. The iPhone 7 also has "3D Touch" interaction that enables users to access even more shortcuts by pressing the screen with higher force.

The configuration was based on the 'out of box' set of applications with the device configured as part of the set-up process. This configuration will vary from user to user but will not make a substantive difference to the common functions available. Although in reality a user would download apps that they want and reconfigure their layout it seems an acceptable compromise to create a benchmark configuration based on the 'out of box' app set. The task sequence explores all the functions available starting from the first home screen with the device awake and unlocked.

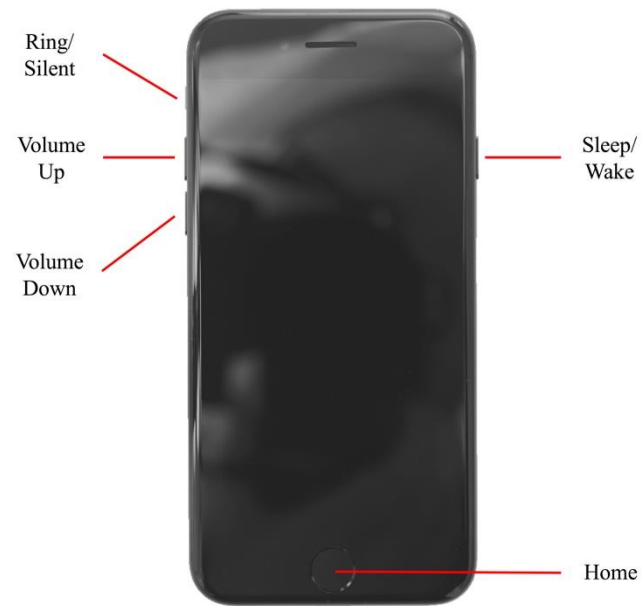


Fig. 1. iPhone 7 showing location of buttons



Fig. 2. iPhone Home screens

Interaction is broken down into the different modalities as follows:

- Buttons.

- Touch.
- Movement of the device.
- Voice.

There are several aspects that require definitions:

- State is the configuration of the phone at a point in time.
- Initial state is the starting point for the analysis. In this case the phone is awake, unlocked and on the first home screen.
- Level refers to the number of states from the initial state that the user navigates through. It is equivalent to the levels in a traditional menu system. Accordingly, the initial state represents the top level or level one.
- Function is defined broadly as an action presented to the user, that either presents a further range of actions or is an action in its own right, such as opening an app or a direct shortcut to an action within an app (e.g. creating a new message). To use the vernacular, a function is “a thing that a user can do”. Such a broad definition is used to see how many of the “things a user can do” are visible. Such a definition is in contrast to more rigorous one, such as Gero’s function-behaviour-state [17], which would have potentially over complicated an already detailed analysis.
- Unique refers to whether the function only appears once within the functions that are considered, in this case functions available from the home screens. Therefore, it is possible that the function may appear elsewhere within iOS and the apps that it supports.
- Visible is defined as any graphical element that is there to indicate the presence of a function.

Where appropriate the terminology, describing the user interface, is based on the iPhone User Guide [10], however terms are changed where greater clarity is required. For example, the difference between a light touch and a more forceful one on the home button is not distinguished completely in the Guide. In this case a double light press is described as a ‘touch’ of the home button which activates the reachability mode as opposed to a double press with greater force which is described as ‘press’, which activates the application switcher.

The analysis was recorded in a spreadsheet showing the functions that are available and the level they are at [18]. This is logically equivalent to a standard WIMP menu hierarchy. A handful of functions are available at all levels and this is noted accordingly. In practice the process was iterative to find the best structure and grouping within the spreadsheet. This was done in conjunction with ‘live’ testing on an iPhone which also revealed further functionality appropriate for analysis.

The analysis should be seen as comprehensive but not exhaustive. This is due in part to the fact any specific configuration does not allow for all options to be available. Also, a full state transition diagram covering every operational situation (e.g. receiving a call when navigating the home screen or the variation in quick action menus depending on previous app usage) would add significant complication. However, for the purposes of this analysis it was deemed that the analysis was sufficiently comprehensive to highlight the issues around visibility.

3 Results

The approach resulted in 622 functions being analysed. As such they are too numerous to display them within the paper but the spreadsheet has been made available [18]. A snapshot example is shown in Table 1 below. This is the analysis of the Mail app. It shows the functions available, which are at level one and two in the hierarchy of functions. The action required to activate the function is recorded as well as a description of the function. The final 3 columns then record if the line item is a function, if the function is unique with the boundary of the home screens function set and if the function is visible at the top level. These columns are then used to count the respective numbers. It should be noted that a function may become visible at a lower level but in terms of discoverability at the top level it is invisible.

Table 1. An Extract from the Analysis Spreadsheet

	A	B	C	D	E	F	G	H	I	J
1		Touch Screen				Action	Function Description	Function	Unique	Visible
2		L1	L2	L3	L4					
3										
108		Mail								
109		App icon				Press	Launch App	Y	Y	Y
110		App icon				Touch & hold →	App Configuration	Y	N	N
111		→ Configure App				Press 'cross'	Delete App	Y	N	N
112		→ Configure App				Drag to new location	Move App	Y	N	N
113		→ Configure App				Drag onto another app	Create Folder	Y	N	N
114		→ Configure App				Drag onto a folder	Add App to Folder	Y	N	N
115		→ Configure App				Drag to dock	Add to dock	Y	N	N
116		→ Configure App				Drag right	Create New Home Screen	Y	N	N
117		App icon				3D press →	Quick Actions Menu	Y	N	N
118		→ Quick Action				Press	New Message	Y	Y	N
119		→ Quick Action				Press	Search	Y	Y	N
120		→ Quick Action				Press	All VIP emails	Y	Y	N
121		→ Quick Action				Press	All inboxes folder	Y	Y	N
122		→ Quick Action				Press	VIP 1 emails	Y	Y	N
123		→ Quick Action				Press	VIP 2 emails	Y	Y	N
124		→ Quick Action				Press	VIP 3 emails	Y	Y	N
125		→ Quick Action				Press	VIP 4 emails	Y	Y	N

Table 2 below provides a numerical summary of the entire analysis for the 4 different modalities considered. A percentage of the functions that are visible is calculated. In addition, a percentage of the visibility of unique functions is calculated separately. This is done because it considers each repeated function, e.g. deleting an app, as a single instance on the basis that if you cannot see app deletion for one app you cannot see it for all the apps. Conversely if you do 'know' it is there it reduces the impact of the lack of visibility of for all instances. It is a debatable point if this is useful, but if used it should be done so in conjunction with the other visibility number that includes repeated functions.

Table 2. A numerical summary of the functions analysed and their visibility

Modality	Number of Functions	Number that are Unique	Number that are Visible	% that are visible	% of unique that are visible
Buttons	23	17	5	22%	29%
Touch	597	266	43	7%	16%
Movement	1	1	0	0%	0%
Microphone	1	1	0	0%	0%
TOTAL	622	285	48	8%	17%

4 Discussion

The bare numbers present what appears to be a stark situation with regards to the high number of functions (622) and the low number of these that are visible (8%) even if repeated functions are removed the visibility figure only rises to 17%. The situation, with regard to the total number of functions, would be increased further as users download additional apps. If one was also to include all the in-app functions, then clearly the number would increase dramatically.

However, the situation is far more nuanced than the bare numbers. The issues around these are as follows:

- The relevance of each functions is not weighted in any way, for example with regard to their importance and frequency of use. As it stands the home screens appropriately prioritise the frequent and important task of launching apps. Secondary functions such as moving or deleting an app are made 'visible' by the 'touch and hold' of the app. This has the big advantage of reducing visual clutter. Indeed, it would be unrealistic to make all the functions visible.
- The numerical analysis does not consider prior knowledge of the users. In other words, their experience of using gestures on a touch device.
- In addition, there is a commercial imperative of offering new features to maintain sales. The needs of different users need to be balanced, in other words there is a function versus complexity trade-off.

Therefore, it can be argued that iOS strikes an appropriate balance between offering a rich set of functions and reducing visual clutter with a focus on the high priority and high

frequency functions. However, the Microsoft Word example cited earlier is a cautionary tale and there is a real concern for novice or older users in particular that these user interfaces become overwhelming. The numerical analysis should not be used as a simple 'good to bad' scoring system but instead it should be used to highlight the overall status of the visibility of a system and to review whether the trade-offs that have been made are appropriate. In addition, support for novice or older users can be looked at, for example providing a simple mode or on screen prompts.

A logical extension of the work is to see if how visibility scores vary across products and how they change over time. Also, it is possible to test if the visibility score is a proxy for overall usability of a product or application. As stated above this simple score does not deal with the nuances of a particular user interface and therefore it is possible that it will not reliably correlate with usability.

5 Limitations

As it stands, the visibility scoring system has only been applied to one operating system and is further limited to the functionality surrounding the home screens. Therefore, without comparing it to other user interfaces and how it may correlate to usability it is not appropriate to cite a score as a good or bad rating for a particular user interface. As it stands the score should be used to inform an evaluation of a user interface and as such is a starting point rather than the end point of an evaluation.

6 Conclusion

The "usability crisis" description has merit, but the situation is nuanced and it is important to look beyond headline statements. There is reason to be concerned that modern post-WIMP interfaces represent a retrograde step with regards to usability. It is reminiscent of the era of the command line "remember and type", but with touch interfaces it is "remember and swipe". It would be a very large stretch to say things are in anyway as difficult as a command line interface but with the increasing reliance on technology it is something that needs to be monitored. Therefore, the ability to methodically assess what is happening with the visibility of user interface functions is important.

The ability to determine a "visibility score" is a contribution to this situation. As it stands the approach has only been applied to one aspect of one product. However, if used appropriately it can help inform both the overall debate and designers who are looking to understand the nature of their design. Further research is merited to compare visibility scores of different elements within products, across products and the evolution of these products over time.

References

1. Nielsen, J.: Noncommand user interfaces. *Communications of the ACM*. 36, 83–99 (1993). doi: 10.1145/255950.153582
2. Gentner, D., Nielsen, J.: The Anti-Mac Interface. *Communications of the ACM*. 39, 70–82 (1996). doi: 10.1145/232014.232032
3. van Dam, A.: Post-WIMP user interfaces. *Communications of the ACM*. 40, 63–67 (1997). doi: 10.1145/253671.253708
4. van Dam, A.: User interfaces: disappearing, dissolving, and evolving. *CACM*. 44, 50–52 (2001). doi: 10.1145/365181.365192
5. Apple Inc.: Q1'16 Earnings Supplemental Material. Apple Inc. (2016).
6. Norman, D.A., Nielsen, J.: Gestural interfaces: a step backward in usability. *interactions*. 17, 46–49 (2010). doi: 10.1145/1836216.1836228
7. Shneiderman, B.: Direct manipulation: A step beyond programming languages. *Computer*. 16, 57–69 (1983). doi: 10.1145/238218.238281
8. Norman, D.A.: Natural user interfaces are not natural. *interactions*. 17, 6–10 (2010). doi: 10.1145/1744161.1744163
9. Caposella, C.: PDC 2005 Keynote with Bill Gates. Presented at the Professional Developers Conference, Los Angeles September 13 (2005).
10. Apple Inc.: iPhone User Guide for iOS 10.2. (2016).
11. English, W.K., Engelbart, D.C., Berman, M.L.: Display-Selection Techniques for Text Manipulation. *Human Factors in Electronics, IEEE Transactions on*. 5–15 (1967). doi: 10.1109/THFE.1967.232994
12. Myers, B.A.: A brief history of human-computer interaction technology. *interactions*. 5, 44–54 (1998). doi: 10.1145/274430.274436
13. Irby, C., Bergsteinsson, L., Moran, T., Newman, W., Tesler, L.: A Methodology for User Interface Design. Xerox Corporation (1977).
14. Smith, D.C., Irby, C., Kimball, R., Verblank, B., Harslem, E.: Designing the Star User Interface. *Byte* (1982).
15. Johnson, J.A., Roberts, T.L., Verplank, W., Smith, D.C., Irby, C.H., Beard, M., Mackey, K.: The Xerox Star: A Retrospective. *Computer*. 22, 11–29 (1989). doi: 10.1109/2.35211
16. Apple celebrates one billion iPhones, <http://www.apple.com/newsroom/2016/07/apple-celebrates-one-billion-iphones.html>, (2016).
17. Gero, J.S.: Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine*. 11, 26 (1990).
18. Hosking, I.: iPhone Visibility Analysis R1, (2017). doi: 10.17863/CAM.7616